Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Курсовая работа по дисциплине

«Базы данных»

Тема: «Разработка базы данных “ Шахматные тактики в онлайн-партиях”»

Выполнил: студент группы ИВТ-22-2Б

Мельников Г. В.

Проверил: Доцент кафедры ИТАС

к.т.н. Петренко А. А.

г. Пермь – 2024

# **Оглавление**

[**Оглавление** 2](#_Toc167340734)

[**Реферат** 3](#_Toc167340735)

[**Введение** 4](#_Toc167340736)

[1. Анализ предметной области 5](#_Toc167340737)

[**1.1. Границы исследования базы данных.** 6](#_Toc167340738)

[2. Проектирование базы данных 7](#_Toc167340739)

[2.1. Концептуальная схема Базы данных 7](#_Toc167340740)

[**2.2 Логическая модель базы данных** 8](#_Toc167340741)

[**2.3 Выбор СУБД** 9](#_Toc167340742)

[**2.4 Построение физической модели базы данных** 11](#_Toc167340743)

[**3. Технология реализации программы** 13](#_Toc167340744)

[**3.1 Выбор языка фреймворка** 13](#_Toc167340745)

[**3.2 Реализация функций и интерфейса** 17](#_Toc167340746)

[**3.4 Тестирование** 22](#_Toc167340747)

[**Заключение** 26](#_Toc167340748)

[**Список литературы** 27](#_Toc167340749)

# **Реферат**

Общий объем: 27 стр. Количество иллюстраций:21. Количество таблиц: 1. Число использованных источников информации: 5.

Ключевые слова: Реляционная база данных, база данных, модели баз данных, Концептуальная модель, Логическая модель, Физическая модель, таблица, СУБД, Шахматные тактики в онлайн-партиях.

Объект исследования: Шахматные тактики в онлайн-партиях.

Предмет исследования: База данных для хранения информации об шахматных тактиках в онлайн-партиях.

Цель работы: разработать базу данных для хранения информации об шахматных тактиках в онлайн-партиях и интерфейс для работы пользователя с БД

Задачи:

1. Провести анализ предметной области.

2.Выбрать подходящее программное обеспечение для реализации программы

3.Разработать концептуальную, логическую и физическую модели данных

4.Разработать базу данных

5.Разработать программу для работы с базой данных

Методы исследования: Анализ, сравнение, эксперимент.

В ходе работы была реализована база данных для хранения информации об шахматных тактиках в онлайн-партиях, а также пользовательский интерфейс для работы с БД

# **Введение**

В современном мире информация стала одним из самых ценных ресурсов. С увеличением объемов данных и сложностью их обработки возникла необходимость в эффективных методах управления и хранения информации. Базы данных и информационные системы играют ключевую роль в этом процессе, предоставляя возможности для структурирования, хранения и быстрого доступа к данным.

Базы данных — это совокупность сведений (о реальных объектах, процессах, событиях или явлениях), относящихся к определенной теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой ее части.

Эта курсовая работа направлена на изучение процесса разработки информационной системы, включая этапы проектирования концептуальной, логической и физической моделей базы данных, а также реализацию и создание пользовательского интерфейса для управления данными.

Создать базу данных шахматных тактик, встречающихся в онлайн-партиях. В партии играют пользователи

Необходимо разработать систему фильтрации для отбора шахматных тактик в онлайн-партиях по критериям:

1. Платформа
2. Дата
3. Контроль времени
4. Дебют
5. Результат (Белые/Чёрные/Ничья)
6. Тактика
7. Страна

1. Анализ предметной области

База данных — это набор упорядоченной, структурированной информации, которая хранится в электронном виде в компьютерной системе.

Управление БД производится с помощью систем управления базами данных (СУБД). СУБД — комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать). Такая система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

Разработка пользовательского интерфейса (программы) потребует использование интегрированной среды разработки, а также структуру, которая станет основой интерфейса.

Для разработки базы данных необходимо использовать СУБД, которая предоставит все необходимые возможности для реализации требований к программе.

Для создания базы данных Шахматные тактики в онлайн-партиях нужно добавлять, изменять и хранить данные, касающиеся шахматных партий.

Так как база данных ориентирована на онлайн-шахматы, то ПАРТИЯ обязана храниться на ПЛАТФОРМЕ и у неё есть ПУТЬ. Партия также содержит такие параметры, как ДАТА, РЕЗУЛЬТАТ (Белые/Чёрные/Ничья), ОБЩЕЕ КОЛИЧЕСТВО ХОДОВ, ВРЕМЯ (время бывает с добавлением и без, к примеру 3:00 – каждому игроку даётся 3 минут без добавления на ход, 3+2 – каждому игроку даётся 3 минуты с добавлением 2 секунд после каждого хода), ДЕБЮТ (определённая комбинация ходов, с которой начинается партия).

В классических шахматах в партии всегда участвуют два ПОЛЬЗОВАТЕЛЯ, у каждого из которых может быть NickName, ИМЯ, ФАМИЛИЯ, ЗВАНИЕ (пользователь может иметь только одно звание или не иметь), СТРАНА, Платформа.

у каждого пользователя в партии есть РЕЙТИНГ, ЦВЕТ фигур (Белые/Чёрные).

В партии может быть множество ТАКТИК (вилка, связка, шах, двойное нападение и т. д.). Тактика начинается с определённого НАЧАЛЬНОГО НОМЕРА ХОДА и длится определённое КОЛИЧЕСТВО ХОДОВ.

Партия ограничена ВРЕМЕННЫМ КОНТРОЛЕМ (Пуля – очень быстро, до 3 минут, Блиц – быстро до 10 минут, Рапид – средне до 30 минут, Классика – медленно, от 30 минут).

## **1.1. Границы исследования базы данных.**

Границы исследования – это факторы и переменные, которые не должны быть включены в исследование. Для данной работы границами исследования являются следующие условности:

1. Для одной платформы все пути к партиям разные, но на нескольких платформах могут быть идентичны
2. В партиях всегда есть ходы, иначе партии не являются значимыми для данной базы данных и не могут в ней находиться
3. на одной платформе NickName различны, но на разных платформах могут совпадать, принадлежать разным пользователям
4. у пользователя всегда есть рейтинг, если пользователь не играл партий, то ему дают начальный рейтинг, к примеру на lichess – 1500
5. на один ход может приходится множество тактик
6. будем считать, что игрок представляет только одну страну или не представляет вообще

**Вывод**

База данных должна предоставлять возможность хранить и обрабатывать необходимые данные. Приложение должно предоставлять возможность пользователю взаимодействовать с базой данных и данными, хранящимися в ней.

2. Проектирование базы данных

Необходимые для разработки базы данных модели, будут созданы с помощью инструмента Draw.IO, так как он является наиболее удобным среди прочих и отвечает всем требованиям, которые необходимы для моделирования БД.[5]

Средством моделирования предметной области на этапе проектирования является модель «сущность-связь». Часто ее называют ER-моделью. В ней моделирование структуры данных предметной области базируется на использовании графических средств - ER-диаграмм. В наглядном виде они представляют связи между сущностями. Основными понятиями ER-диаграммы являются сущность, атрибут, связь. Сущность представляет собой объект, информация о котором хранится в базе данных. Сущность имеет экземпляры, отличающиеся друг от друга значениями атрибутов и допускающие однозначную идентификацию. Атрибут - это свойство сущности. Атрибут, который уникальным образом идентифицирует экземпляры сущности, называется ключом.

2.1. Концептуальная схема Базы данных

Концептуальная схема – определяющая представление базы данных, единое для всех ее приложений и не зависящее от используемого в СУБД представления данных в среде хранения. Концептуальная модель базы данных построена по нотации Питера Чена, как самой наглядной из существующих.

Ход работы:

1. Создание таблиц, отражающую необходимую информацию о базе данных
2. Создание связей между таблицами, которые позволят пользователю корректно взаимодействовать с нужными данными.

Концептуальная модель представлена на Рисунке 1.

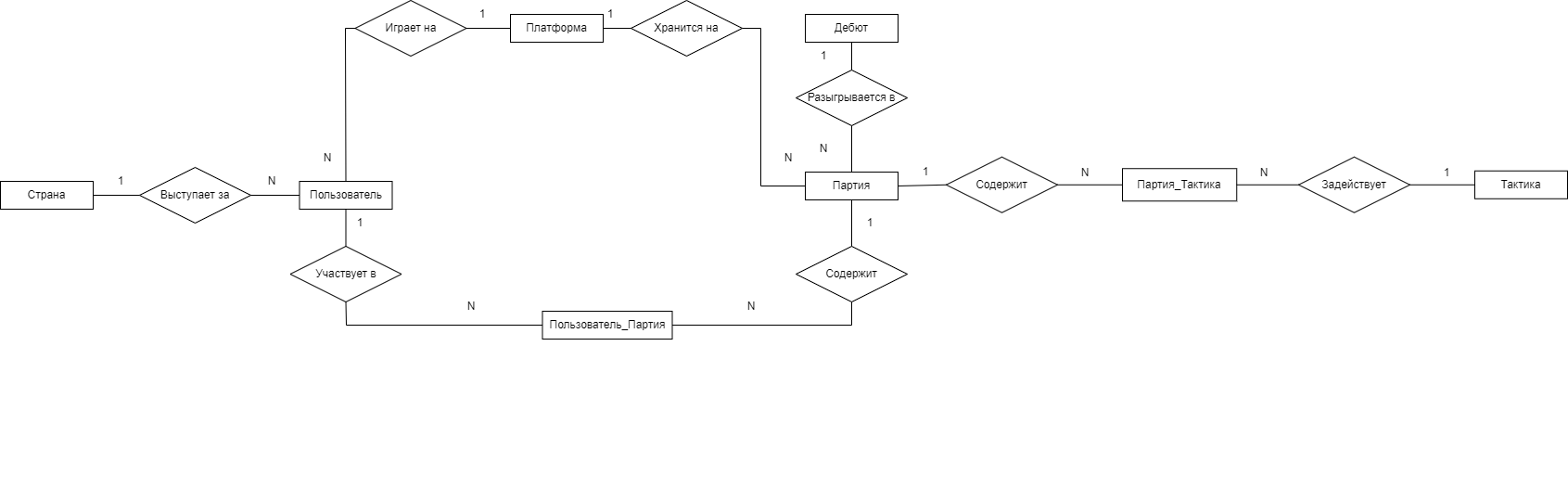


Рисунок 1 - Концептуальная модель базы данных

Концептуальная модель базы данных содержит 8 сущностей и 8 связей, корректно отображающих информацию базы данных.

## **2.2 Логическая модель базы данных**

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД.

Во время разработки логической модели данных определяются такие понятия, как первичный ключ - в реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа, внешний ключ - столбец или комбинация столбцов, значения которых соответствуют Первичному ключу в другой таблице

Логическая модель была построена по нотации Мартина, как наиболее наглядная и легкоусвояемая нотация из имеющихся нотаций на сегодняшний день.

Ход работы:

1. Определение первичных и внешних ключей для каждой сущности базы данных.
2. Построение связей непосредственно между соответствующими ключами сущностей.

Логическая модель изображена на Рисунке 2.

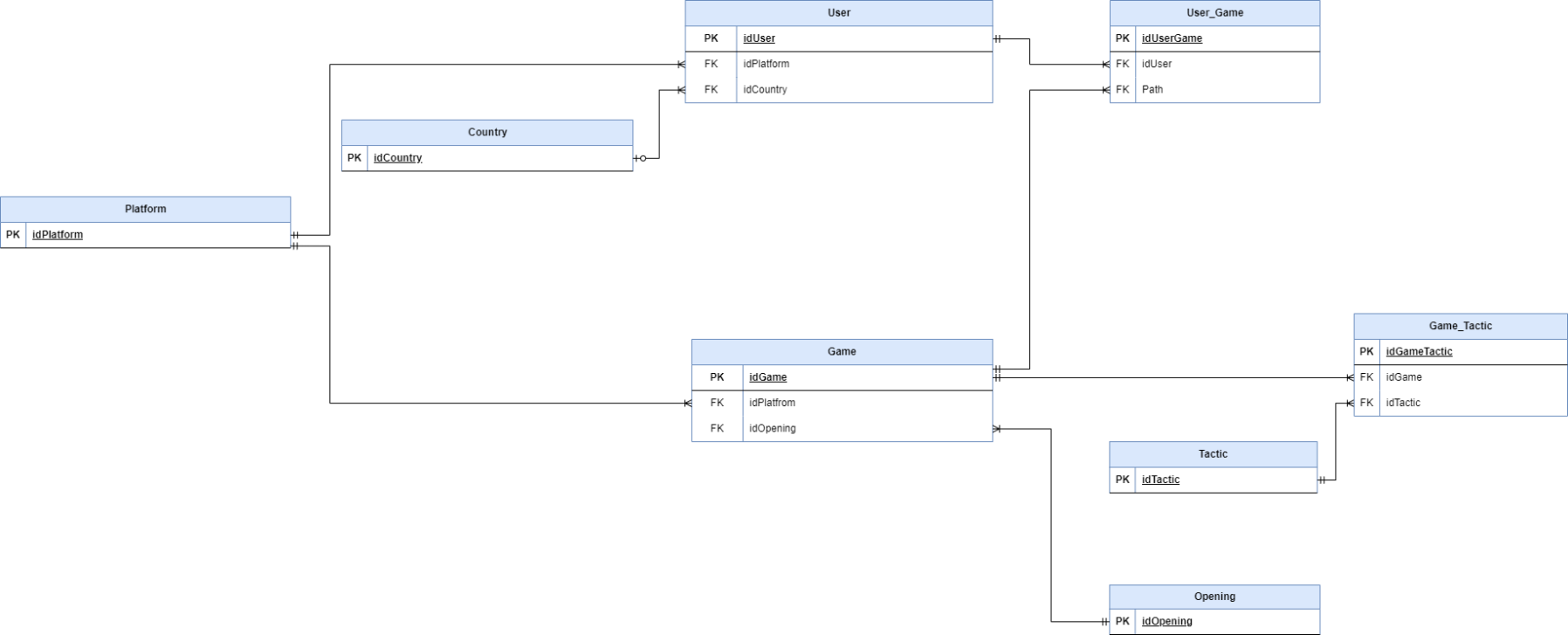


Рисунок 2 - Логическая модель базы данных

## **2.3 Выбор СУБД**

После составления логической модели, на ее основании была выбрана СУБД для перехода на дальнейшие пункты работы. Для выбора подходящего СУБД был проведен сравнительный анализ двух СУБД (Таблица 3).

Таблица 1 - Сравнение различных СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **MySQL** | **PostgreSQL** | **Microsoft SQL Server** |
| Скорость | Быстрая работа с данными, особенно при чтении | Хорошая скорость как при чтении, так и при записи | Быстрая работа с данными, особенно при записи |
| Масштабирование | Легко добавлять новые серверы, делить данные на части | Можно добавлять новые серверы, но сложнее настроить | Хорошее вертикальное масштабирование, но дорого |
| Сообщество и помощь | Много пользователей, много инструкций и помощи | Активное сообщество, много документации | Платная поддержка, официальные ресурсы |
| Лицензия | Бесплатно (GPL), есть платные версии | Бесплатно, нет скрытых затрат | Дорогостоящие лицензии и поддержка |
| Простота использования | Легко устанавливается и настраивается | Требует больше знаний для настройки | Простой интерфейс, но сложные настройки |
| Безопасность | Много встроенных средств безопасности | Очень гибкие настройки безопасности | Высокий уровень безопасности, но сложные настройки |
| Стоимость | Бесплатно, есть платные версии для бизнеса | Бесплатно, нет скрытых затрат | Дорогостоящие лицензии и поддержка |

Исходя из вышеописанной сравнительной таблицы, принято решение по использованию MySQL как основного вида СУБД в дальнейшей работе, так как данный инструмент является наиболее простым в понимании и использовании.

## **2.4 Построение физической модели базы данных**

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных. Физическая модель базы данных строится на основе логической и содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей.

Ход работы:

1. Добавление не ключевых полей в таблицы.
2. Определение типов данных для каждого поля.
3. Определение автоинкрементных и ненулевых полей.
4. Определение значений по умолчанию
5. Определение ограничений
6. Определение триггеров
7. Определение представлений

Во время разработки физической модели данных было принято решение использовать следующие типы данных:

1. INT – целочисленный тип данных.
2. NVARCHAR(x) - Строковый тип содержащий символы Unicode. Длина строки <= x.
3. DATE - тип данных, содержащий информацию о дате в формате “ГГГГ-ММ-ДД”

Физическая модель была построена по нотации Мартина.

Физическая модель изображена на Рисунке 3.

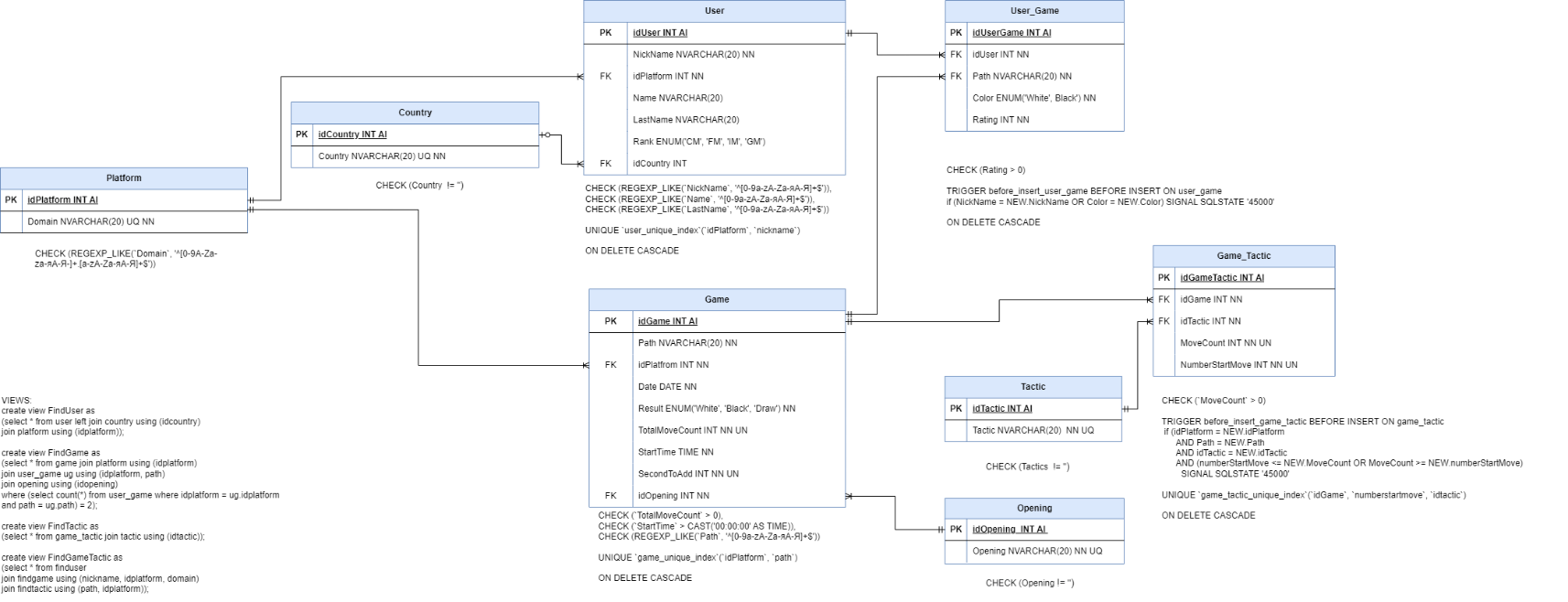


Рисунок 3 - Физическая модель базы данных

**Вывод**

База данных будет реализована с помощью СУБД MySQL. Для отражения всех необходимых данных и запросов будет создано 8 сущностей и 8 связей.

# **3. Технология реализации программы**

## **3.1 Выбор языка фреймворка**

Для создания приложения выбран фреймворк WPF.

Преимуществами WPF является:

1. Богатый пользовательский интерфейс

WPF предоставляет мощные средства для создания сложных и привлекательных пользовательских интерфейсов. Это включает поддержку:

Графики и визуальных эффектов: WPF поддерживает 2D и 3D графику, а также множество анимаций и визуальных эффектов.

Сложных макетов: Возможность создавать сложные интерфейсы с помощью различных контейнеров макета, таких как Grid, StackPanel, DockPanel и т.д.

Темизации и стилизации: Легкость применения стилей и тем к элементам управления для создания уникального внешнего вида приложений.

2. Разделение логики и представления

WPF поддерживает архитектуру Model-View-ViewModel (MVVM), которая способствует разделению логики приложения и его представления, что упрощает поддержку и тестирование кода.

Data Binding: WPF предоставляет мощные возможности привязки данных, что позволяет легко связывать данные из модели с элементами управления.

Command Binding: Позволяет связывать действия пользовательского интерфейса с логикой приложения, улучшая чистоту и тестируемость кода.

3. Поддержка мультимедиа

WPF включает встроенную поддержку для работы с мультимедийными элементами:

Видео и аудио: Легко интегрировать видео и аудио контент в приложения.

Изображения и векторная графика: Поддержка различных форматов изображений и возможность работы с векторной графикой.

4. Интеграция с другими технологиями

WPF может быть интегрирован с другими технологиями и библиотеками, что делает его гибким для различных сценариев использования:

Интеграция с Windows Forms: Возможность встраивания элементов управления Windows Forms в WPF и наоборот.

Интеграция с DirectX: Позволяет использовать мощь DirectX для создания высокопроизводительных графических приложений.

5. Поддержка современных стандартов

WPF поддерживает современные технологии и стандарты, что делает его актуальным для разработки новых приложений:

XAML: Использование XAML (Extensible Application Markup Language) для декларативного определения пользовательского интерфейса.

.NET: Полная интеграция с платформой .NET, что позволяет использовать все преимущества этой экосистемы.

6. Высокая производительность

WPF использует аппаратное ускорение для рендеринга графики, что обеспечивает высокую производительность и плавность интерфейсов.

Аппаратное ускорение: Использование возможностей графических процессоров (GPU) для рендеринга.

Асинхронная обработка: Поддержка асинхронных операций, что позволяет создавать отзывчивые приложения.

7. Мощные инструменты разработки

Microsoft предоставляет мощные инструменты для разработки и дизайна WPF приложений:

Visual Studio: Интегрированная среда разработки с мощными возможностями отладки и дизайна.

Blend for Visual Studio: Инструмент для дизайнеров, который облегчает создание и настройку визуальных интерфейсов.

8. Поддержка международных стандартов

WPF поддерживает локализацию и глобализацию, что позволяет создавать приложения для международных рынков.

9. Обширная библиотека компонентов

Существует множество готовых компонентов и библиотек для WPF, что ускоряет разработку и позволяет легко добавлять новые функции в приложения.

В целом, WPF предлагает мощные инструменты для создания современных, высокопроизводительных и привлекательных настольных приложений для Windows, делая процесс разработки более эффективным и приятным.

10. Интеграция с базами данных

Привязка данных (Data Binding)

WPF обладает мощной системой привязки данных, которая позволяет легко связать пользовательский интерфейс с данными из базы данных. Это включает:

Простая привязка: Связывание элементов интерфейса с источниками данных с помощью XAML или C# кода.

Двусторонняя привязка (Two-way Binding): Обеспечивает синхронизацию данных между моделью и интерфейсом, что удобно для создания форм ввода данных.

Поддержка ORM

WPF хорошо работает с различными объектно-реляционными мапперами (ORM), такими как Entity Framework. Это позволяет разработчикам легко взаимодействовать с базами данных, используя объектно-ориентированный подход.

Entity Framework: Позволяет автоматически генерировать модели данных из базы данных и управлять данными с помощью LINQ-запросов.

Работа с различными типами данных

WPF поддерживает работу с данными различных типов и форматов:

Коллекции данных: Поддержка работы с коллекциями данных (например, ObservableCollection), которые автоматически обновляют интерфейс при изменении данных.

Data Templates: Позволяет легко настраивать отображение данных в интерфейсе, используя шаблоны данных.

Инструменты для работы с данными

WPF предоставляет множество элементов управления для отображения данных, таких как DataGrid, ListView, TreeView и другие. Эти элементы позволяют создавать удобные и функциональные интерфейсы для работы с данными.

DataGrid: Мощный элемент управления для отображения табличных данных с поддержкой сортировки, фильтрации и редактирования.

ListView и TreeView: Подходят для отображения иерархических данных и списков.

Обработка данных на стороне клиента

С WPF вы можете обрабатывать данные на стороне клиента, что помогает снизить нагрузку на серверную часть и улучшить производительность приложения.

Кэширование данных: Возможность кэширования данных на клиенте для быстрого доступа и улучшения отзывчивости интерфейса.

Валидация данных: Встроенные механизмы для валидации данных на клиенте перед отправкой их в базу данных.

## **3.2 Реализация функций и интерфейса**

При входе в приложение пользователя встречает окно авторизации. После попытки входы выполняется попытка подключение к бд.

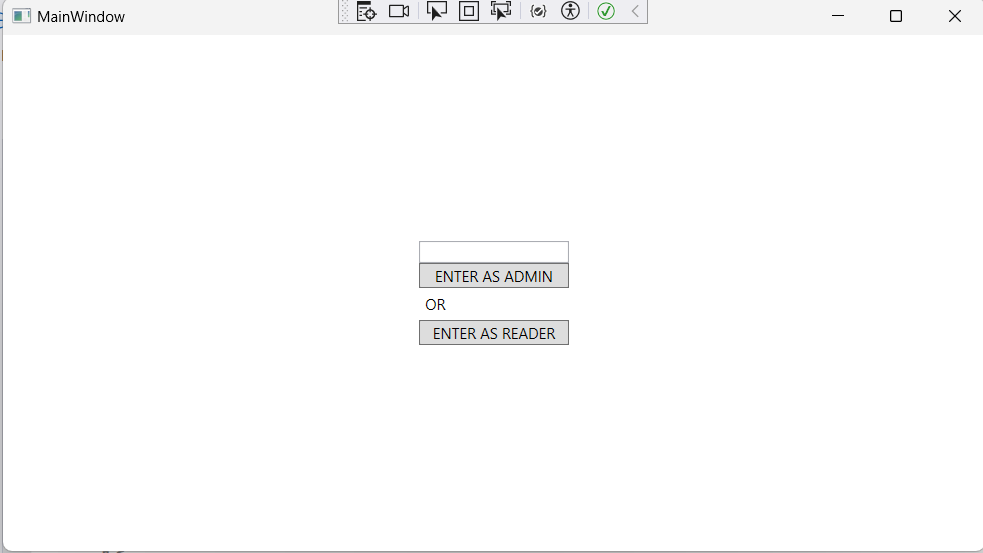


Рисунок 4 – Вход в приложение

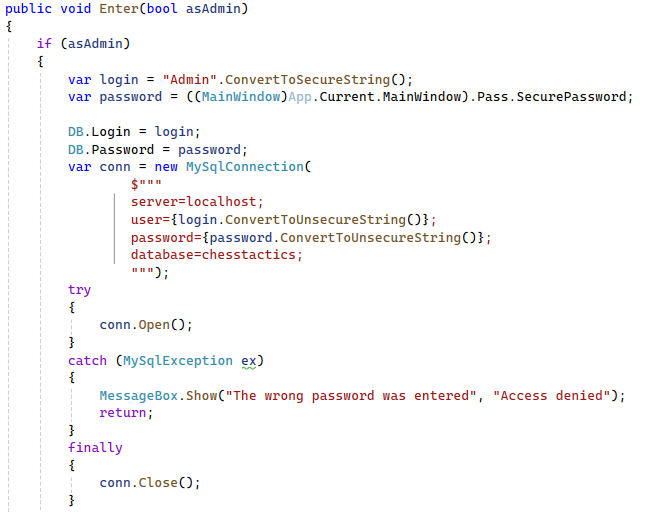


Рисунок 5 – код авторизации

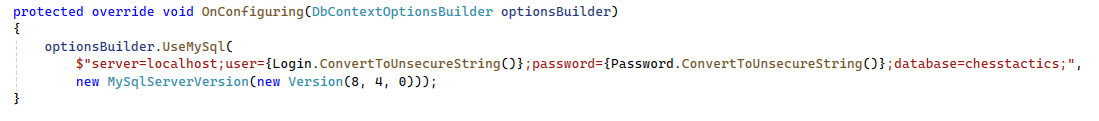


Рисунок 6 – код подключение к бд

Если пользователь вошёл в приложение как администратор, то у него появится панель редактировани

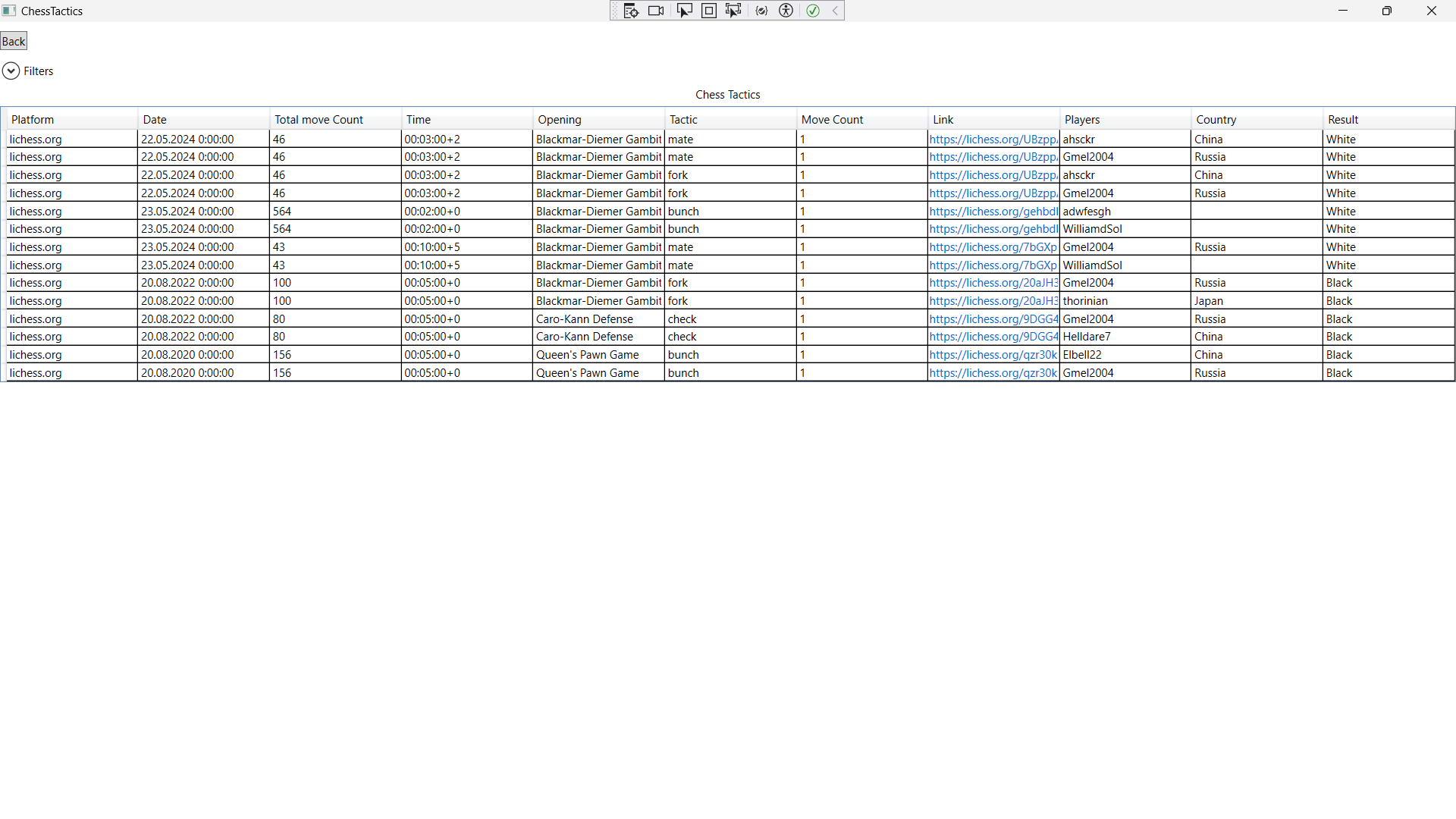


Рисунок 8 – вход в приложение с правами просмотра

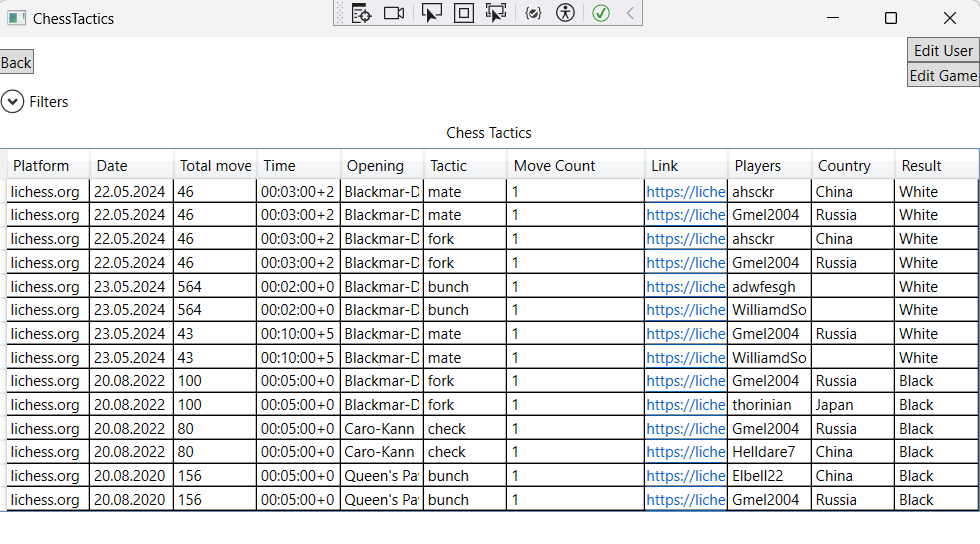


Рисунок 9 – вход в приложение с правами админа

На рисунке 10 изображена партия,котораая отображается после надатия на ссылку в таблице



Рисунок 10 - партия

На рисунке 11 показана фильтрация, можно отобрать тактики по критериям

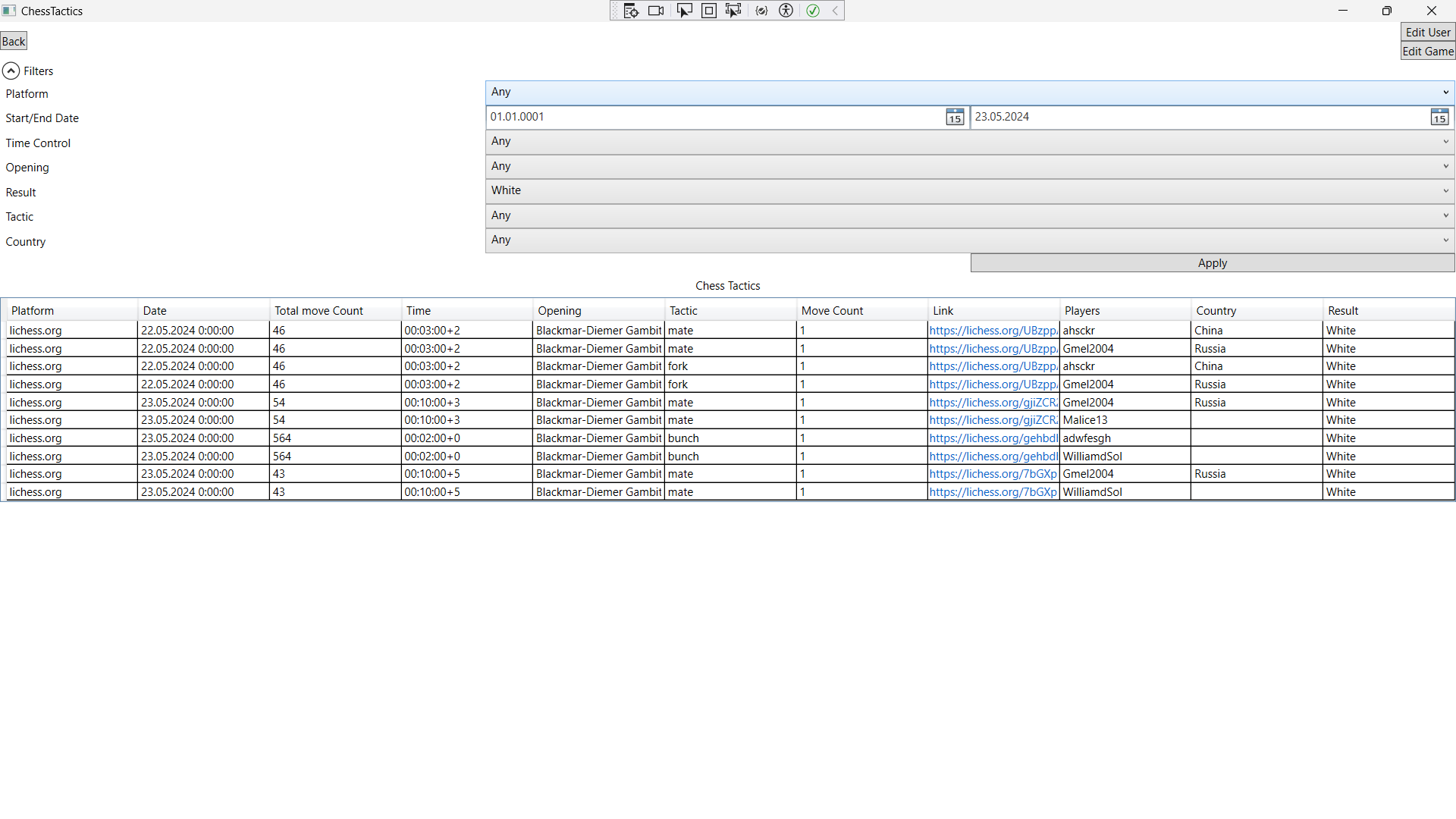


Рисунок 11 - фильтрация

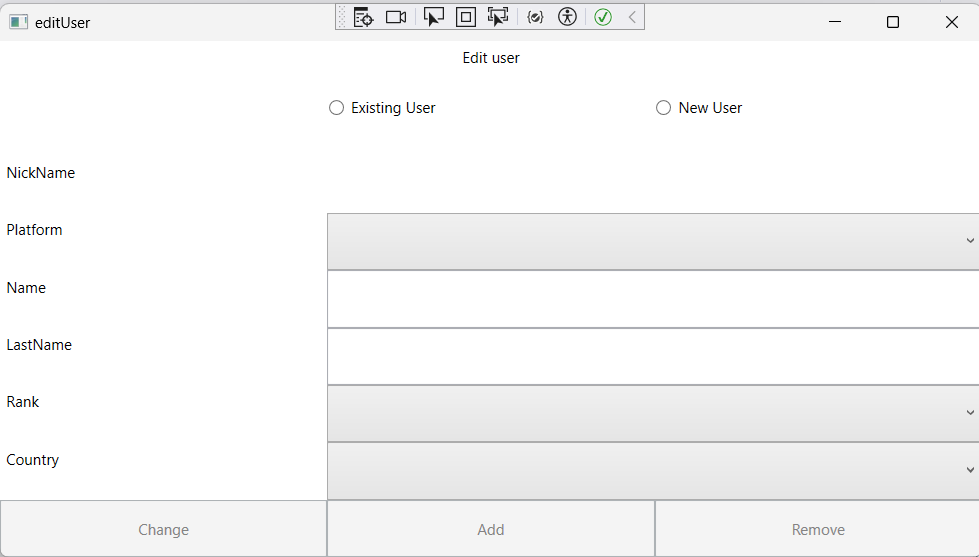


Рисунок 12 – редактирование пользователя

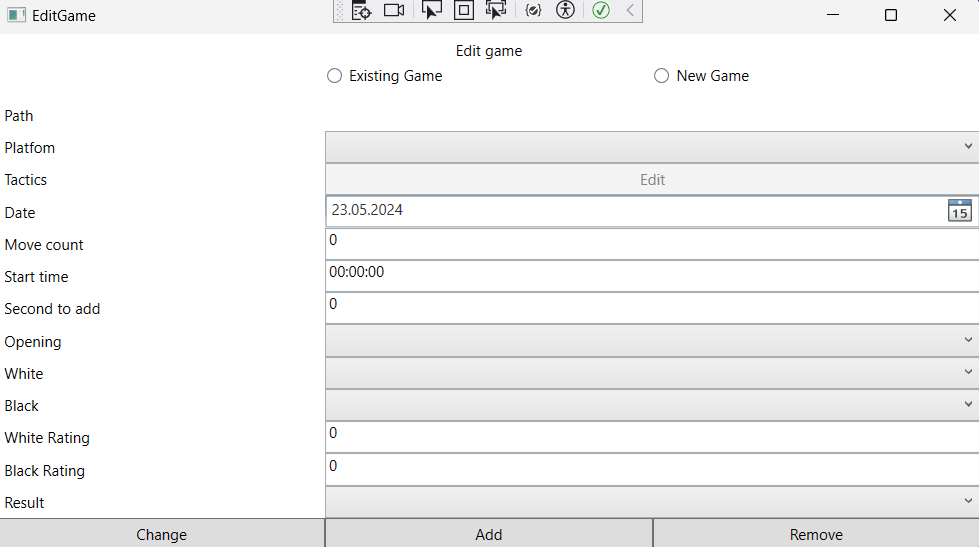


Рисунок 13 – редактирование игры

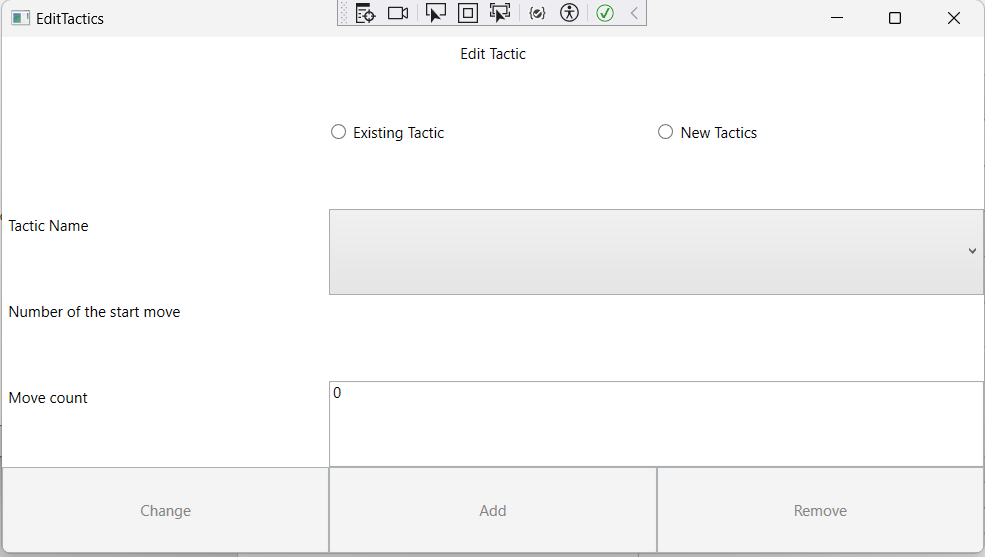


Рисунок 14 – редактирование тактики

## **3.4 Тестирование**

Для тестирования приложения выполним следующие действия:

* + - 1. Создадим пользователя
      2. Создадим партию
      3. Добавим тактику в партию
      4. Убедимся, что тактика появилась в таблице
      5. Попробуем перейти по ссылке

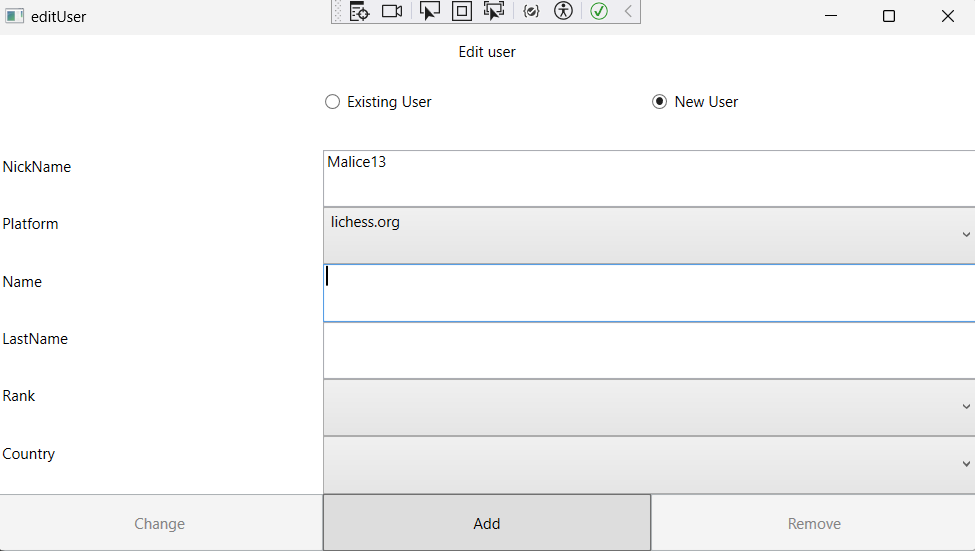


Рисунок 15 – создание пользователя

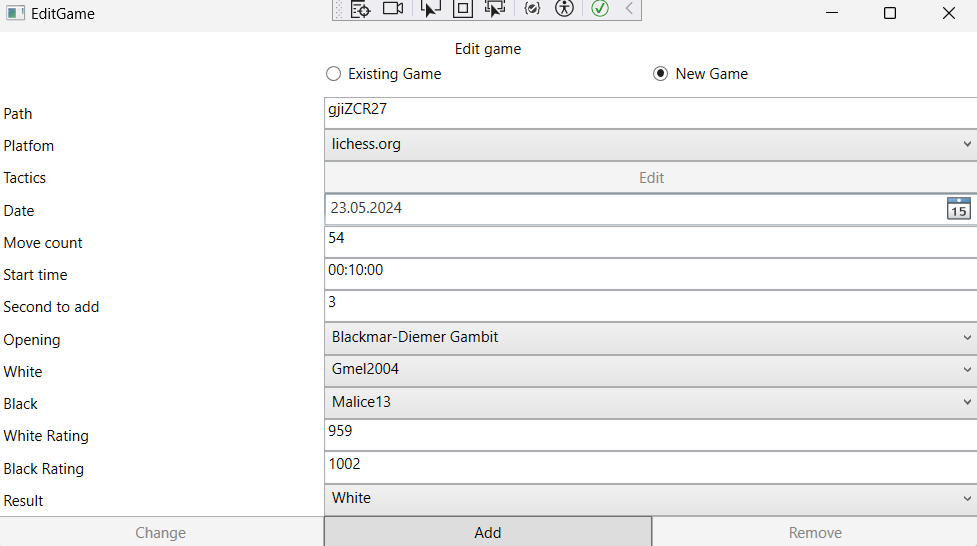
****

Рисунок 16 – создание игры

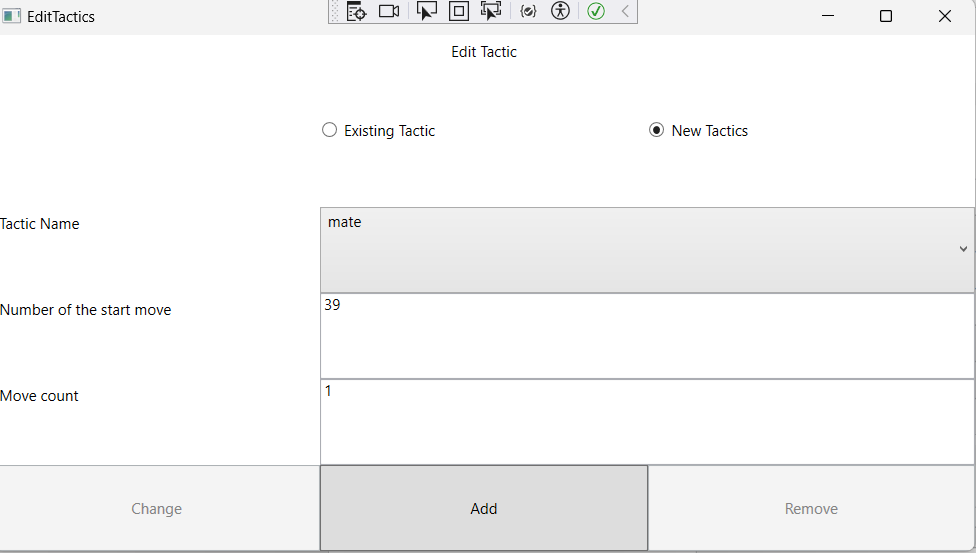
****

Рисунок 17 – добавление тактики

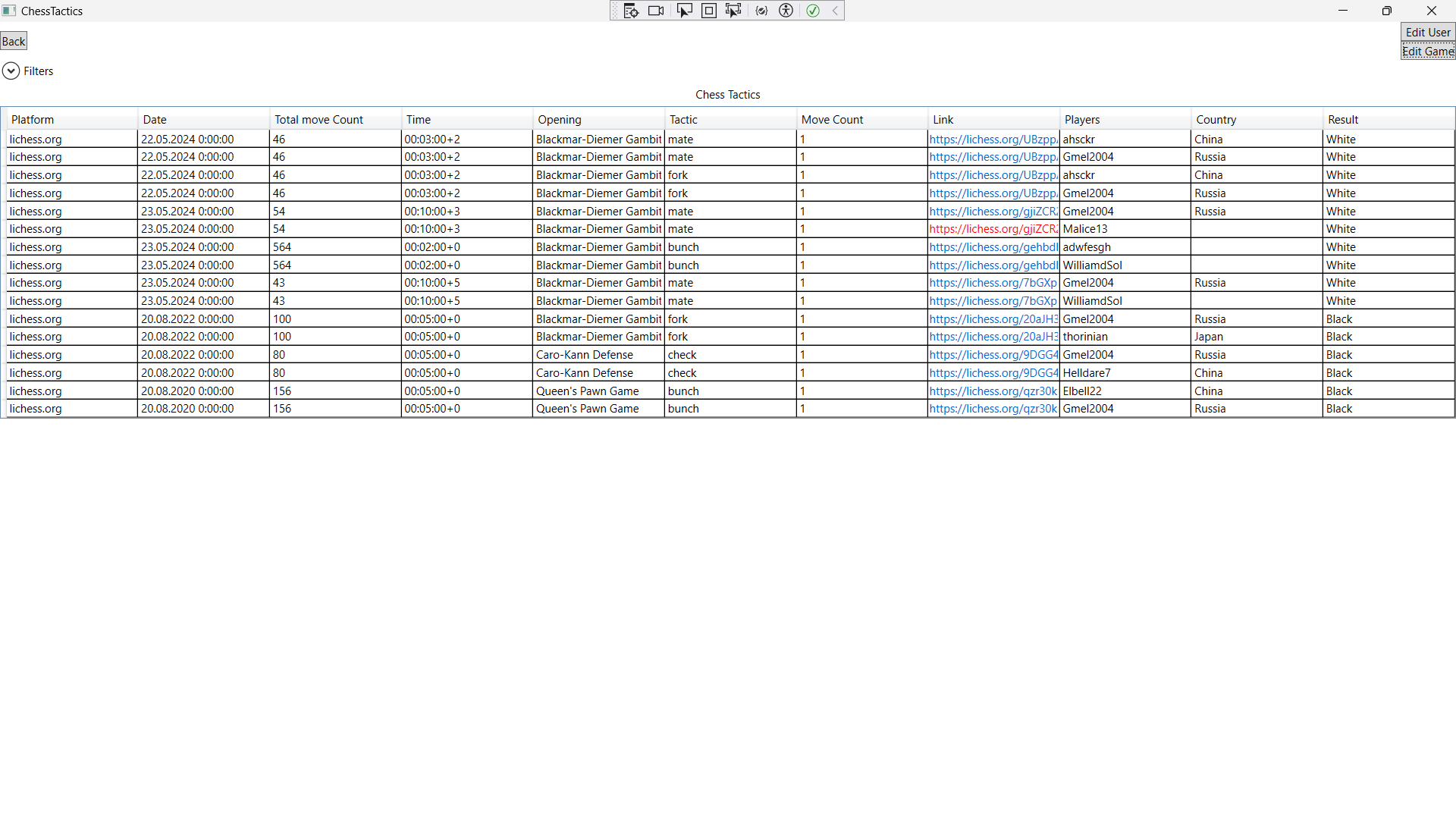
****

Рисунок 18 – проверка добавления записей в таблицу

****

Рисунок 19 – переход по ссылке

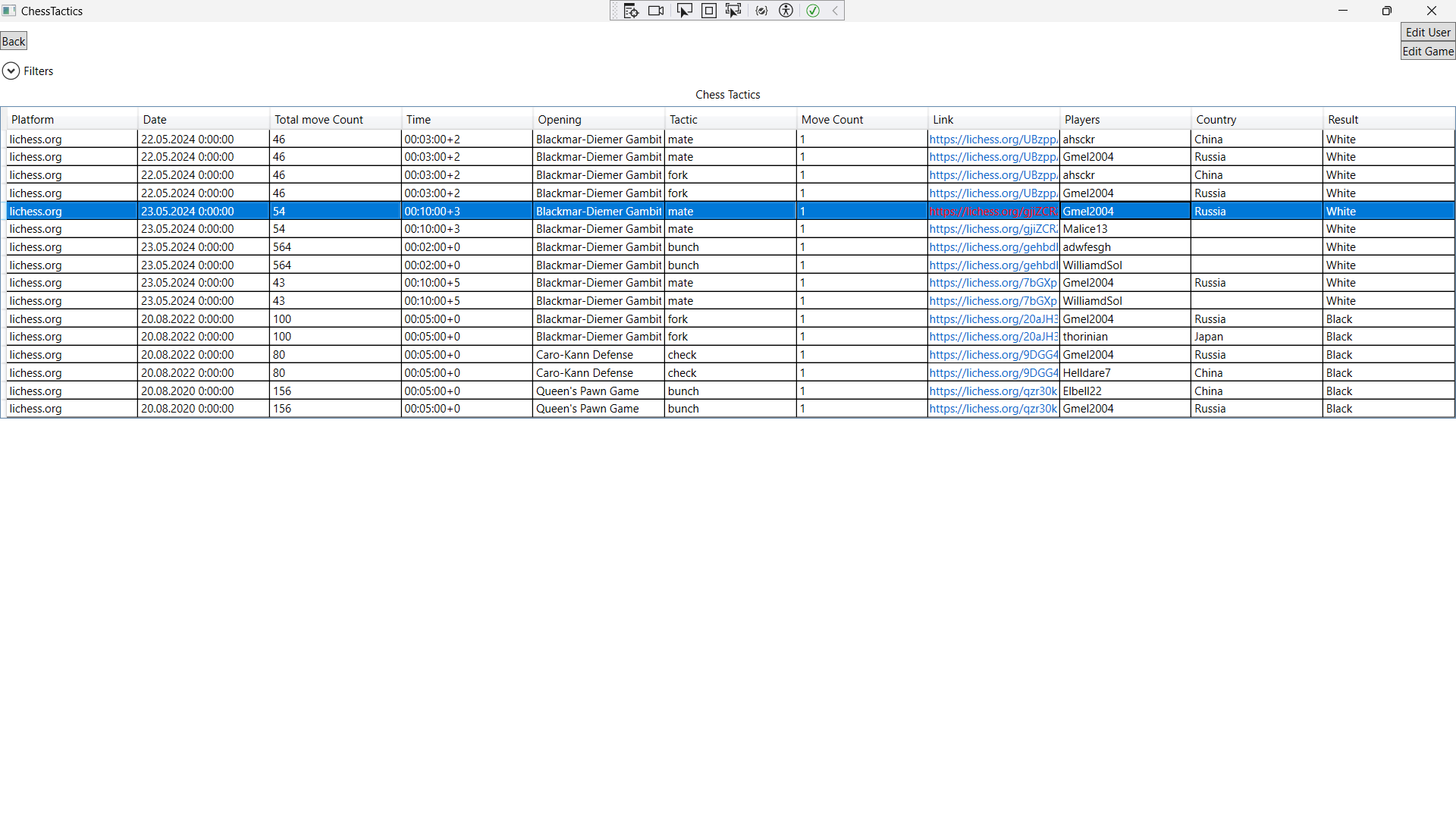
****

Рисунок 20 – проверка добавления записей в таблицу

****

Рисунок 20 – переход по ссылка

# **Заключение**

В ходе выполнения курсовой работы была создана базы данных для шахматных тактик в онлайн-партиях. Была также разработана программа для удобного пользования элементами СУБД пользователя.

Дальнейшей перспективой работы является расширение платформ, для сбора шахматные тактик в онлайн-партиях с различных сайтов.

# **Список литературы**

1.   Балалаев М.А. Проектирование баз данных: Методические указания по выполнению курсового проекта по дисциплине «Базы данных» / М.А. Балалаев; ДВГУПС. .     Каф. «Системы автоматизированного проектирования». Хабаровск: Изд - во ДВГУПС, 2007. - 30 с. .     Гурвиц Г.А. Мюгозой Ассезз 2007. Разработка приложений на реальном примере/ Г.А. Гурвиц. - СПб.: БХВ-Петербург, 2007. - 672 с .

2.  "Pro C# 8 with .NET Core 3: Foundational Principles and Practices in Programming" by Andrew Troelsen and Philip Japikse

3.  Прайс Джейсон Oracle Database 11g: SQL. Операторы SQL и программы PL/SQL; Лори - , 2012. - 660 c. .

4. Астахова И. Ф., Мельников В. М., Толстобров А. П., Фертиков В. В. СУБД. Язык SQL в примерах и задачах; ФИЗМАТЛИТ - Москва, 2012. - 168 c.  
5 Инструмент для онлайн-моделирования Draw.io [Электронный ресурс] - <https://app.diagrams.net/> - (Дата обращения 22.04.2024)